

---

# Il Tempo e i Calcolatori

---

# Sommario

---

- Caratterizzazione del tempo
  - I Ruoli del Tempo nei Sistemi di Calcolo
  - Componenti Temporal
  - Errori di Temporizzazione
  - Requisiti dei Temporizzatori
  - Tipologie di Temporizzazioni Attive
  - Tecniche Realizzative dei Timer
-

---

# Caratterizzazione del Tempo

---

# Caratterizzazione del Tempo

---

- Modello del tempo

- ▶ Modello concettuale/continuo:  $t$  (tempo assoluto)

- Tempo come contenitore dei fenomeni o come prodotto dai fenomeni

- ▶ Modello informatico/discreto:  $T$  (tempo locale)

- Associato a un fenomeno orologio (*clock*) che determina la **risoluzione temporale**

- $utl \Rightarrow$  unità di tempo locale [sec]

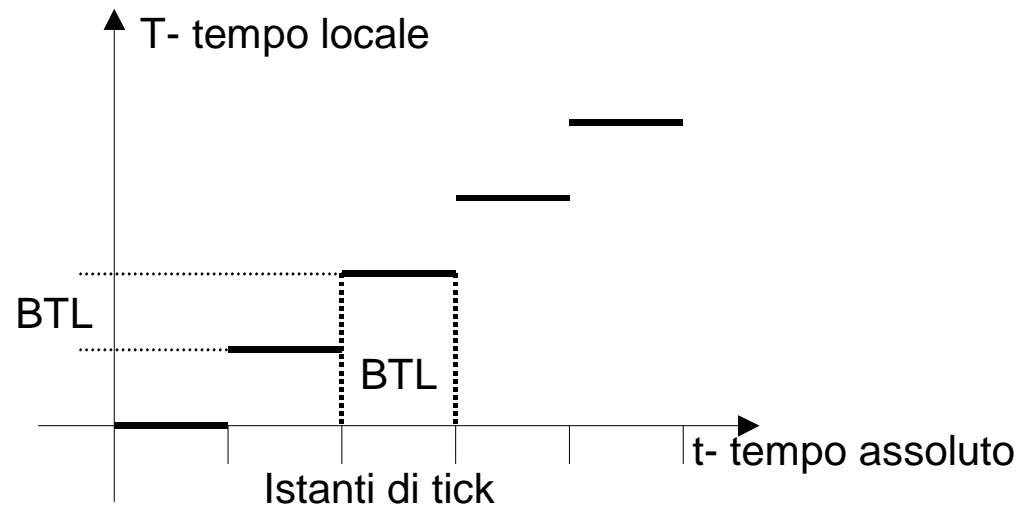
- Risoluzione temporale locale: è una proprietà della rappresentazione locale del tempo (esempio: risoluzione di 1 msec)

- $BTL = K * utl \Rightarrow$  base tempo locale [sec]

- Periodo di aggiornamento dalla variabile *orologio* (esempio: granularità di 55 unità da 1 msec cioè un aggiornamento ogni 18,2 volte al secondo)
-

# Caratterizzazione del Tempo

- ▶ Operativamente per ogni CPU si esiste una variabile "orologio" che gode della proprietà di incrementare di BTL (ogni BTL) il proprio valore quando si verificano particolari eventi (tick)
- ▶ Il comportamento dell'orologio è espresso da una funzione nel dominio del tempo  $T = f(t)$  definita come  $T = n * BTL$  ( $n \in \mathbb{N}$ ) per  $n * k * utl \leq t < (n+1) * k * utl$ 
  - Nota: con un orologio di N bit  $T = (n \bmod 2^N) * BTL$



# Caratterizzazione del Tempo: Errori

---

- Gli errori che si commettono nell'adottare come misura del tempo una variabile orologio di tempo locale sono scomponibili in due contributi
    - ▶ Errori dovuti al fenomeno assunto come orologio
      - Precisione
    - ▶ Errori dovuti alla granularità
      - Quantizzazione
-

# Errori del Fenomeno

---

- Orologio preciso

- ▶  $t(\text{tick}[n]) - t(\text{tick}[n-1]) = \text{BTL} + e[n]$  con  $e[n]=0$

- ▶ Caratteristiche di  $e[n]$

- Errore sistematico:  $E[ e[n] ] \neq 0$

- Orologio poco accurato

- » Il fenomeno assunto come sorgente presenta una frequenza non corretta

- Errore a valore atteso nullo:  $E[ e[n] ] = 0$

- Orologio poco ripetibile

- » Irregolarità compensante (jitter)

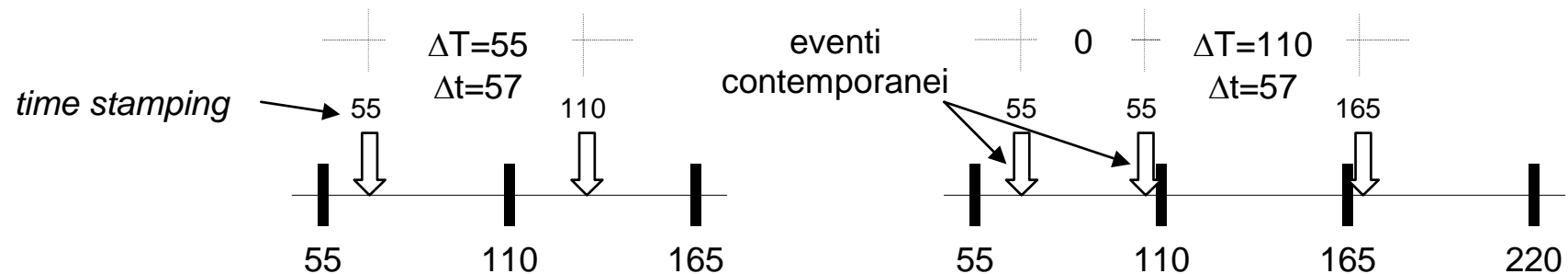
- Orologio Corretto al tempo assoluto  $t$

- ▶  $\text{Abs}(T - t) < \text{BTL}$

---

# Errori di Granularità

- Ordinamento e contemporaneità
  - ▶ Eventi che si verificano tra due *tick* consecutivi non sono ordinabili temporalmente e sono da considerare contemporanei nel tempo locale pur distando di un tempo assoluto  $\Delta t \leq BTL$
- Misure di intervalli
  - ▶ Eventi distanti  $\Delta t$  nel tempo assoluto distano  $\Delta T$  nel tempo locale
    - $\lfloor \Delta t / BTL \rfloor * BTL \leq \Delta T \leq \lceil \Delta t / BTL \rceil * BTL$





---

# I Ruoli del Tempo nei Sistemi di Calcolo

---

# I Ruoli del Tempo nei Sistemi di Calcolo

---

- Generatore di eventi (Flusso di Eventi Temporali)
    - ▶ Il tempo come sorgente di stimoli impliciti cioè non prodotti da fenomeni del mondo esterno
      - Individua momenti per attivare
        - Campionamenti di stati
        - Emissione di informazioni
        - Aggiornamento di "data ora"
    - ▶ Ruolo attivo
      - Uso di interruzioni per notificare gli eventi temporali e fare attivare le azioni corrispondenti
        - Granularità grossolana di 0.5 .. 100 ms
-

# I Ruoli del Tempo nei Sistemi di Calcolo

---

- Grandezza di stato
    - ▶ Valore da acquisire in particolari momenti
      - Per misure di intervalli temporali da una origine arbitraria
        - Es. Durata di un intervallo temporale tra due eventi
      - Per associare l'informazione relativa al tempo corrente
        - Es. *Time-stamping*
    - ▶ Ruolo passivo
      - Come variabile da leggere nell'istante desiderato
        - Granularità Fine di 1...100  $\mu$ s
-

# I Ruoli del Tempo nei Sistemi di Calcolo

---

- Temporizzatore per dispositivi HW
  - ▶ Generatore di eventi a cui sono sensibili dei dispositivi elettronici e solo in subordine l'eventuale attivazione di elaborazioni interne al calcolatore
  - ▶ Ruolo attivo
    - Produce azioni su HW ed eventi tramite interruzioni
      - Granularità da molto Fine a Grossolana: dipende dal dispositivo



---

# Componenti Temporali

---

# Componenti Temporali

---

- Il comportamento temporale dei sistemi di calcolo è caratterizzato da
    - ▶ Ritardi tra stimoli e risposte delle azioni sporadiche
    - ▶ Periodo e irregolarità (jitter) delle azioni periodiche
  - Le componenti temporali che intervengono nella esecuzione delle elaborazioni sono
    - ▶ Latenza
    - ▶ Tempo di overhead
    - ▶ Tempo di comunicazione
    - ▶ Tempo di elaborazione netto
-

# Componenti Temporali

---

- Latenza

- ▶ Intervallo di tempo durante il quale l'esecuzione dell'azione stimolata non può iniziare o non può proseguire
    - È dovuta alla scarsità di risorse (tipicamente la CPU)
      - Le risorse sono dedicate ad altre attività
  - ▶ Latenze tipiche
    - Latenza di risposta a richieste di interruzione
    - Latenza di attivazione di un processo *ready*
  - ▶ In genere, è di difficile valutazione perché dipende dal contesto di elementi che si contendono l'uso delle risorse e dalle politiche di assegnazione di queste
    - Latenze medie, massime
      - Estremi inferiore e superiore
-

# Componenti Temporali

---

- Tempo di *overhead*
    - ▶ Tempo che i meccanismi di base (HW e soprattutto SW/SO) spendono per la gestione delle azioni applicative
      - Spesso tale tempo è accorpato al *tempo di elaborazione netto*
    - ▶ Efficienza dei meccanismi di base
      - Rapporto tra tempo dedicato alle azioni utili e tempo totale
        - Il tempo di overhead è necessario ma non produttivo
    - ▶ Il tempo di overhead è relativamente stabile e poco dipendente dal contesto
    - ▶ In molti casi si accettano elevati overhead (bassa efficienza) in cambio di prestazioni più ricche, più robuste e più prevedibili da parte del SO
-



# Componenti Temporali

---

- Tempo di comunicazione

- ▶ Una componente *additiva* del ritardo globale è costituita dai tempi richiesti dalle comunicazioni di informazioni

- Dipende dal mezzo trasmissivo e dalla tecnica di comunicazione utilizzata

- Può essere un collo di bottiglia

- In molti casi sono disponibili risorse dedicate alla comunicazione che possono lavorare in parallelo alla CPU

---

# Componenti Temporal

---

- Tempo di comunicazione

- ▶ Il tempo di comunicazione è costituito da due componenti principali
    - Tempo impiegato per il protocollo (overhead)
      - Meccanismi di accesso al mezzo trasmissivo e sincronizzazione
      - Meccanismi di gestione ed instradamento
      - Meccanismi per rilievo e correzione di errori
    - Tempo impiegato per il trasferimento
      - Questa componente diventa preponderante se le informazioni utili sono organizzate in "blocchi" di dimensioni sostanziose
  - ▶ Questi tempi sono prevedibili con buona approssimazione
-

# Componenti Temporali

---

- Tempo di elaborazione netto (componente effettiva)
    - ▶ È il tempo dedicato dalla CPU ad eseguire le istruzioni dell'algoritmo *effettivo*
      - Le altre componenti *additive* del tempo (latenza, tempo di overhead e tempo di comunicazione) sono fondamentali per completare l'elaborazione
    - ▶ Il tempo netto di elaborazione è di difficile valutazione
      - Dipendente dai costrutti di programmazione utilizzati (costrutti condizionali, cicli, ecc.) e dalle condizioni al contorno
      - Misurati empiricamente tramite opportuni tool
        - Es. Profiler
-

---

# Errori di Temporizzazione

---

# Errori di Temporizzazione

---

- I tempi coinvolti nell'esecuzione delle azioni, analizzati nelle sezioni precedenti, hanno un impatto sulle precisioni delle operazioni temporali
  - Accenniamo ad alcuni aspetti generali in relazione a tre casi principali
    - ▶ Rilievo del tempo assoluto di un evento
    - ▶ Misura di un intervallo
    - ▶ Esecuzione a tempo prefissato di un azione
    - ▶ N.B.
      - Trascureremo gli errori di precisione e di quantizzazione dell'orologio adottato che sono comunque aggiuntivi
-

# Errori di Temporizzazione

---

- Rilievo del tempo assoluto di un evento
    - ▶ La concatenazione dei tempi tra il verificarsi di un evento e l'acquisizione del tempo relativo è la seguente
      - $T_a$ : latenza del rilievo dell'evento (var.)
      - $T_b$ : Tempo di esecuzione del riconoscimento dell'evento (cost.)
        - Le componenti  $T_a$  e  $T_b$  si hanno solo per eventi esterni che devono essere recepiti a controllo di programma o ad interrupt
      - $T_c$ : Latenza dell'operazione di lettura del valore del tempo (var.)
        - La componente  $T_c$  può essere ridotta a zero se si rende non interrompibile la sequenza di operazioni di lettura del tempo
          - » Ad esempio disabilitando gli interrupt
      - $T_d$ : Tempo di esecuzione della lettura del valore tempo (cost.)
        - La componente  $T_d$  è ineliminabile, ma piuttosto costante e breve
-

# Errori di Temporizzazione

---

- Rilievo del tempo assoluto di un evento
    - ▶ L'errore temporale **minimo** è quindi pari a
      - Per eventi esterni  $T_b + T_d$
      - Per eventi interni  $T_d$
    - ▶ L'errore temporale **massimo** è
      - Per eventi esterni  $T_{amax} + T_b + T_{cmax} + T_d$
      - Per eventi interni  $T_{cmax} + T_d$
    - ▶ Tali errori temporali sono sempre per eccesso
-

# Errori di Temporizzazione

---

- Misura dell'intervallo tra due eventi
    - ▶ La misura di un intervallo di tempo viene generalmente effettuata rilevando il tempo assoluto dell'evento di inizio, quello dell'evento di fine e calcolandone la differenza
    - ▶ Gli errori dipendono solo dalle componenti variabili che sono tipicamente costituite dalle latenze
      - Le componenti temporali costanti contribuiscono a formare il ritardo della disponibilità della misura, ma non ne inficiano il valore
    - ▶ Gli errori di misura massimi per eccesso e per difetto (cioè in modulo) sono quindi pari alla somma delle latenze massime ( $T_{amax}$  e  $T_{cmax}$  visti precedentemente) dato che le latenze minime sono nulle
-



# Errori di Temporizzazione

---

- Esecuzione a tempo prefissato di un'azione
    - ▶ Concatenazione dei tempi tra evento temporale e completamento dell'azione
      - $T_a$ : Latenza del rilievo dell'evento temporale
      - $T_b$ : Tempo di esecuzione del riconoscimento dell'evento temporale
        - $T_a$  e  $T_b$  sono relativi ai meccanismi di *percezione* del tempo
          - » Interrupt o controllo di programma
      - $T_c$ : Latenza dell'azione
        - $T_c=0$  se si rende non interrompibile la sequenza di operazioni
          - » Quando ciò non è possibile questa latenza può diventare dominante
      - $T_d$ : Tempo di esecuzione dell'azione
        - La componente  $T_d$  è ineliminabile
    - ▶ Gli errori sono sempre per eccesso (ritardo) e vanno da
      - un minimo di  $T_b + T_d$
      - a un massimo di  $T_{amax} + T_b + T_{cmax} + T_d$
-

---

# Requisiti dei Temporizzatori

---

# Requisiti dei Temporizzatori

---

- I requisiti dei temporizzatori influenzano la scelta dei meccanismi da cui dipendono le varie componenti di errore di temporizzazione e delle granularità da cui dipendono gli errori di quantizzazione
  - È buona regola che la granularità dei valori della grandezza sia dello stesso ordine di grandezza delle incertezze su tali valori
  - Si usano le valutazioni sulle incertezze nei diversi ruoli del tempo per scegliere le corrispondenti granularità
    - ▶ Attivazioni di processi e loro elaborazioni (attivo)
      - Ordine delle centinaia di istruzioni macchina
      - Granularità dei tempi di “risveglio” di attività è dell'ordine del msec
    - ▶ Lettura del tempo da associare ad un evento (passivo)
      - Ordine alcune istruzioni macchina
      - Granularità di misura dell'ordine del  $\mu\text{sec}$
    - ▶ Attivazione diretta di dispositivi circuitali
      - Ordine del nsec
      - Granularità, se le specifiche lo richiedono, inferiore ai 100 nsec
-

---

# Tipologie di Temporizzazioni Attive

---

# Tipologie di Temporizzazioni Attive

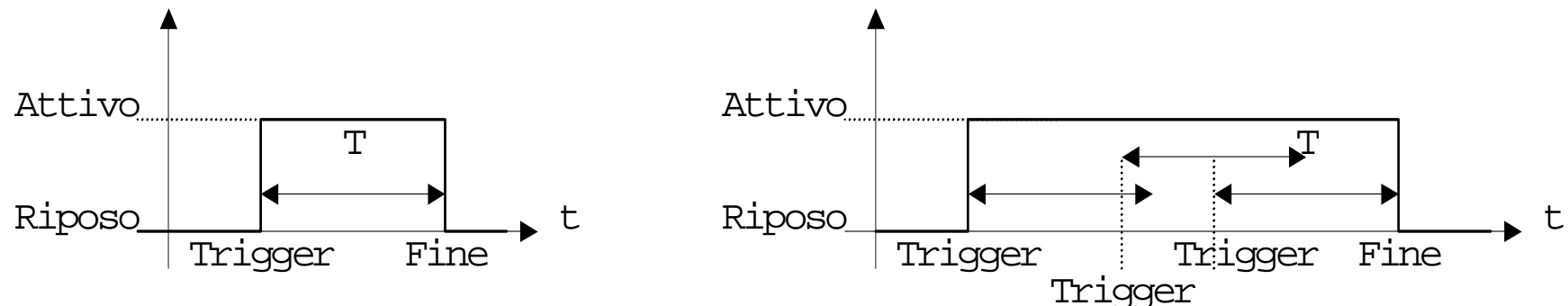
---

- Le temporizzazioni attive e generatrici di eventi sono soprattutto utilizzate per
    - ▶ Produrre stati temporizzati durante i quali eseguire attività
    - ▶ Produrre eventi temporali in cui eseguire azioni
  - Una tipica classificazione distingue diversi tipi di timer
    - ▶ Timer a *durata*
    - ▶ Timer a *ritardo*
    - ▶ Timer *ciclici*
-

# Tipologie di Temporizzazioni Attive

- Temporizzazione a *durata*

- ▶ È una temporizzazione di tipo *monostabile* attivata da un evento *trigger* e conclusa dall'evento di *fine attivazione* dopo l'intervallo di tempo prefissato
- ▶ Sono detti *re-triggerabili* (prorogabili) i temporizzatori in cui il conteggio del tempo riparte da zero ad ogni nuovo evento di *trigger* prolungando così l'attivazione

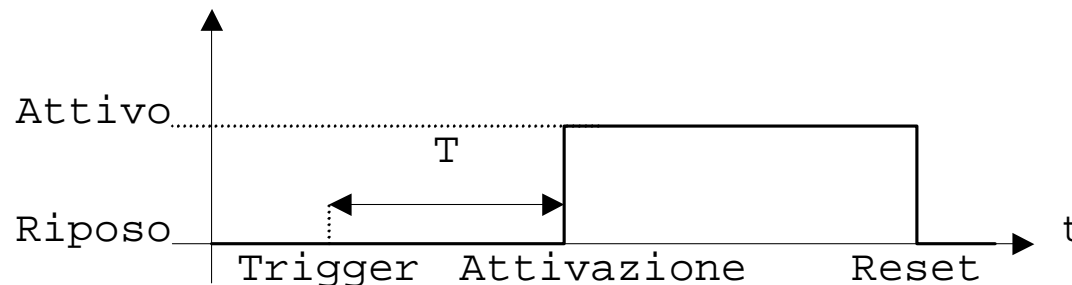


T=valore di temporizzazione

# Tipologie di Temporizzazioni Attive

---

- Temporizzazione a *ritardo*
  - ▶ L'evento *trigger* attiva solo il conteggio dell'intervallo specificato, al termine del quale si verifica l'evento *attivazione*
  - ▶ In genere esiste un comando *reset* per la disattivazione con il ritorno nello stato di riposo
    - Se l'evento di *reset* si verifica prima della scadenza di T non si ha l'attivazione



T=valore di temporizzazione

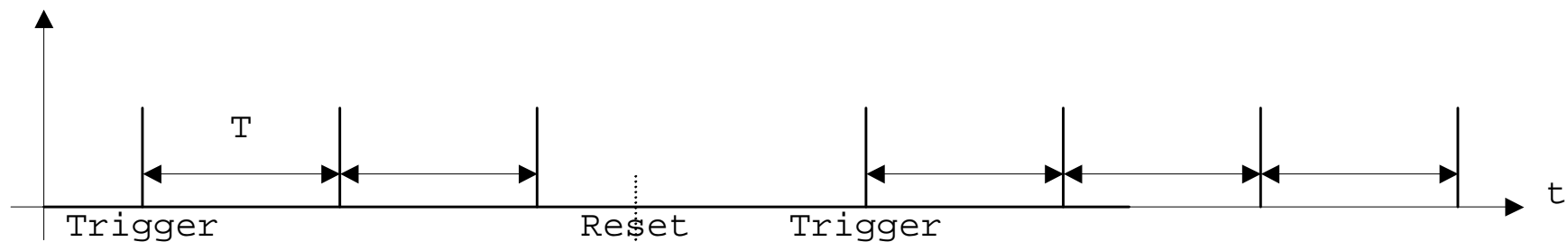
---

# Tipologie di Temporizzazioni Attive

---

- Temporizzazione *ciclica*

- ▶ Dopo l'evento di *trigger* si producono eventi con cadenza regolare e di periodo temporale prefissato
- ▶ Il *reset* arresta il funzionamento
  - Il funzionamento riprende solo dopo un nuovo evento di *trigger*



$T$ =valore di temporizzazione

---



---

# Tecniche Realizzative dei Timer

---

# Tecniche Realizzative dei Timer

---

- Realizzazioni HW: circuiti *Calendar/Clock*
    - ▶ Circuiti integrati con funzione di orologio e calendario
      - Non hanno funzione di temporizzatori a causa della risoluzione/granularità temporale (1 sec)
        - Utilizzano tipicamente un oscillatore a 32768 Hz
    - ▶ Contengono contatori necessari per tener conto di secondi, minuti, ore, giorno della settimana, mese, anno
    - ▶ Tipicamente, la lettura avviene serialmente per ridurre il pin-out
      - Operazione lenta (centinaia di microsecondi)
    - ▶ Sono spesso corredati di batterie tampone
-

# Tecniche Realizzative dei Timer

---

- Realizzazioni minimali HW/SW
    - ▶ Semplici orologi basati sulla generazione di segnali a cadenze regolari (*tick*) forniti al circuito di generazione degli *interrupt*
      - Real-Time Clock (RTC)
        - Oscillatore quarzato con divisore di frequenza (*prescaler*)
    - ▶ Tutte le funzioni sono svolte a livello SW dalla routine di servizio delle interruzioni dei *tick* di RTC
      - La routine aggiorna una variabile di tipo *ora/data*
        - Utilizzate per operazioni di time-stamping e per misura di intervalli
      - La routine verifica se uno o più intervalli temporali relativi ai processi scade ed effettua le eventuali transizioni di stato dei processi interessati
-

# Tecniche Realizzative dei Timer

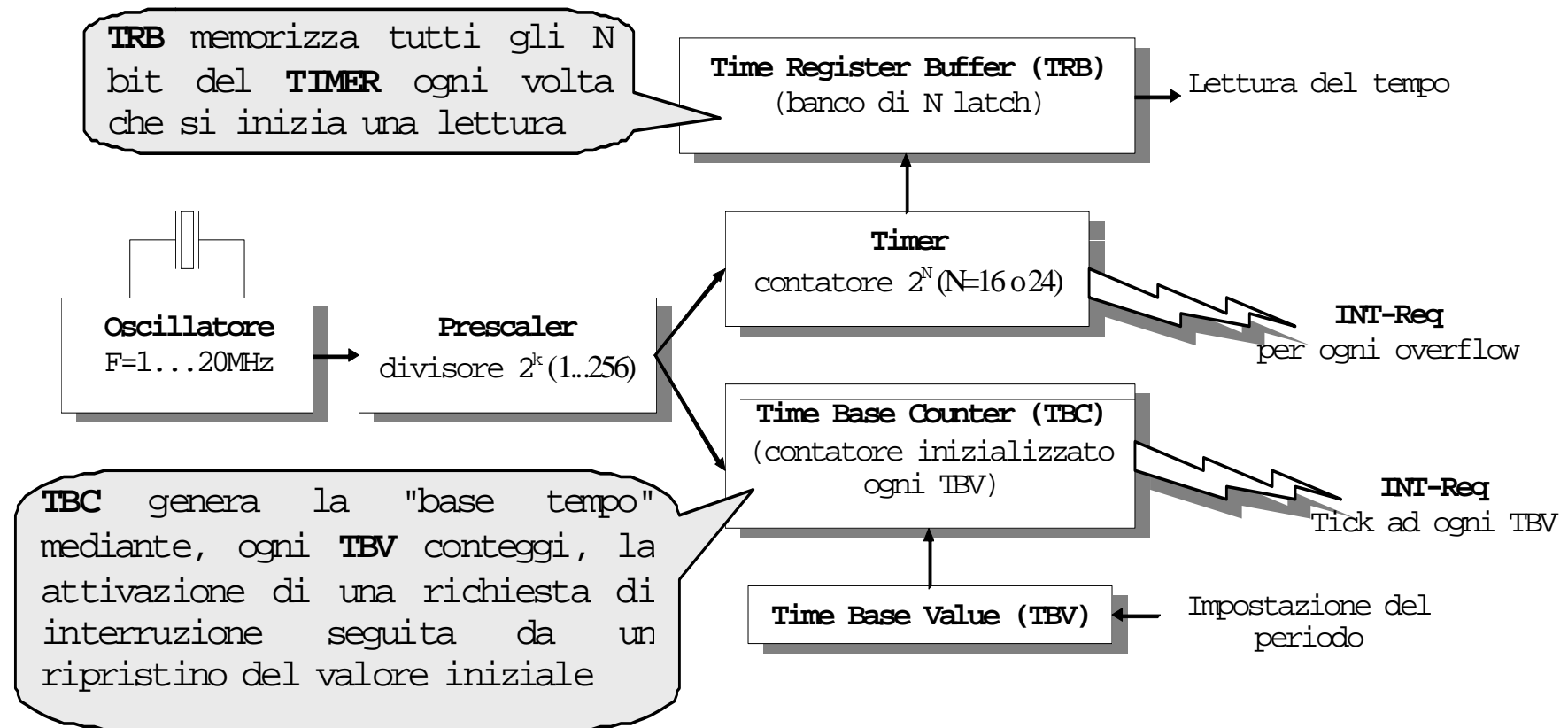
---

- Realizzazioni minimali HW/SW
    - ▶ Con questa tecnica la risoluzione e le granularità di *percezione* e di *misura* coincidono e corrispondono al periodo  $T_{ck}$  dei tick
    - ▶ La frequenza dei tick è scelta come compromesso tra una fine granularità temporale ed un overhead accettabile
      - L'overhead corrisponde al tempo di esecuzione medio della routine di servizio
        - Dovrebbe essere inferiore ad un decimo di  $T_{ck}$
      - Valori tipici di  $T_{ck}$ 
        - 1..10 msec per real-time
        - 100 msec per applicazioni senza particolari requisiti temporali
-

# Tecniche Realizzative dei Timer

- Realizzazioni tipiche HW/SW

- Obiettivo principale: varie granularità adatte ai diversi scopi



# Tecniche Realizzative dei Timer

---

- Realizzazioni tipiche HW/SW
    - ▶ La precisione di questi orologi e la stessa dell'oscillatore
  
    - ▶ Timer
      - Risoluzione (TIR) = Granularità (TIR)
        - $2^k * \text{PeriodoOscillatore} = 2^k / \text{FreqOsc} [\text{sec}]$  (tipicamente 1...100  $\mu\text{sec}$ )
      - Periodo
        - $\text{TIP} = 2^N * \text{TIG} [\text{sec}]$  (tipicamente da 64 msec a circa 16 sec)
          - »  $\text{uti}=\text{TIG}$  e  $\text{BTL}=1$
  
    - ▶ Time Base
      - Risoluzione (TBR)
        - $2^k * \text{PeriodoOscillatore} = 2^k / \text{FreqOsc} [\text{sec}]$  (tipicamente 1...100  $\mu\text{sec}$ )
      - Granularità (TBG) = Periodo (TBP)
        - $\text{TBV} * \text{TBR} [\text{sec}]$  (tipicamente 1...50 msec)
          - »  $\text{uti}=\text{TIG}$  e  $\text{BTL}=\text{TBV}$
-

# Tecniche Realizzative dei Timer

---

- Realizzazioni tipiche HW/SW

- ▶ Meccanismi di ripristino di TBC a TBV

- HW

- Ricaricamento automatico contestuale con la generazione della richiesta di interruzione
      - Basato su un valore impostato in TBV via SW in fase di inizializzazione

- SW

- Ricaricamento tramite routine di risposta all'interruzione
        - Il conteggio prosegue portando all'accumulo di eventuali ritardi: il nuovo periodo non inizia dal termine del periodo precedente
        - Una soluzione è quella di adottare un contatore a decremento e di sommare il TBV (o dualmente)
-

# Tecniche Realizzative dei Timer

---

- Realizzazioni tipiche HW/SW

- ▶ Conclusioni

- La gestione temporale dei processi è effettuata dalla routine di risposta ai tick di TBC
    - L'aggiornamento di data e ora può essere effettuato, per gli usi a "grana grossa", dalla risposta ai tick di TBC o agli overflow di TIMER
    - Le misure di intervalli di tempo più raffinate utilizzeranno i valori di conteggio leggibili dal contatore TIMER
      - Poichè il TIMER ha periodo TIP, questo è il più lungo intervallo di tempo misurabile senza ambiguità
        - » Misure di tempo di intervalli più lunghi possono ottenersi, utilizzando opportuni accorgimenti, utilizzando la variabile data-ora aggiornata a SW per risolvere le ambiguità del tempo ciclico ricavato da TIMER





# Tecniche Realizzative dei Timer

---

- Realizzazioni complesse

- ▶ Processori speciali

- Nei processori per applicazioni di tempo reale stretto sono spesso presenti meccanismi HW che consentono di effettuare operazioni relative al tempo con granularità raggiungibili solo a livello HW, grazie a latenze e tempi di esecuzione trascurabili rispetto al SW
  - Es. Microcontroller Intel 80C196KB

- ▶ Coprocessori di Temporizzazione

- Per raggiungere gli obiettivi di elevate risoluzioni e precisioni con minimo overhead per la CPU si può affidare ad un *coprocessore di temporizzazione* la gestione di funzioni relative al tempo
    - Dispositivi ausiliari di gestione del tempo sono necessari se si vogliono utilizzare per applicazioni real-time processori "*general purpose*" che sono potenti ma poco dotati di funzioni per il tempo
-

# Tecniche Realizzative dei Timer

---

- Realizzazioni SW

- ▶ In molte applicazioni si ha la necessità di gestire diverse temporizzazioni, anche tra loro concomitanti
- ▶ Spesso non è conveniente adottare un temporizzatore HW per ogni temporizzazione, ma risulta efficace e flessibile una soluzione SW
  - Ad esempio, è possibile definire temporizzatori SW, gestiti dalla routine di risposta alle interruzioni di un unico temporizzatore HW
    - Questo approccio è adottato, sia pure con diverse modalità, da quasi tutti i nuclei di sistemi operativi real-time
    - I processi applicativi chiedono al SO, con apposite primitive, che gli venga assegnato un temporizzatore associato ad una certa funzione

